User Manual

Steps To Create Form in Vienna Advantage

Version 1.0

1 Table of Contents

1. Introduction 3
2. Process To Create A Form 3
2.1 Step 1 3
2.2 Step 2 4
2.3 Step 3 6
2.4 Step 4 8
3. Resize Design of User Control 9
4. Maintaining Logs on The form 9
5. Get And Set Context 10
6. How To Show Message In Pop-Up Window 10
7. Get Text According to culture 10
8. Working with Database 11
9 Entry in the Form Window 12
10 Bind Form with Menu 13
11 Give Access to Form 15
12 Run The Form 16

1. Introduction

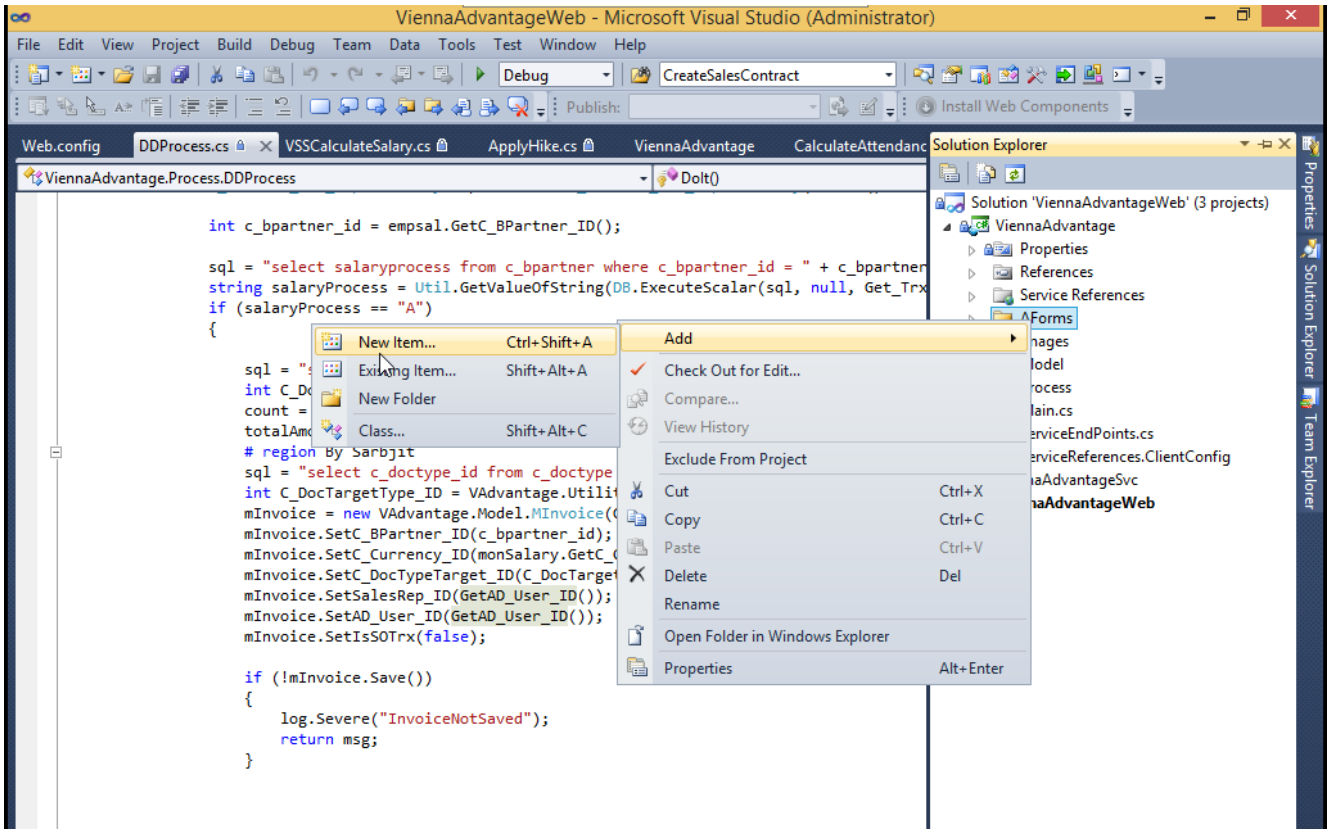
A form is a formatted User Interface containing blank fields that users can fill in with data, Dropdowns, Checkboxes and other controls. To store user data into our database or Update data we need forms. It is a kind of a window which is customized according to the user requirement and a bit of coding involved as well which should be done to create a form.

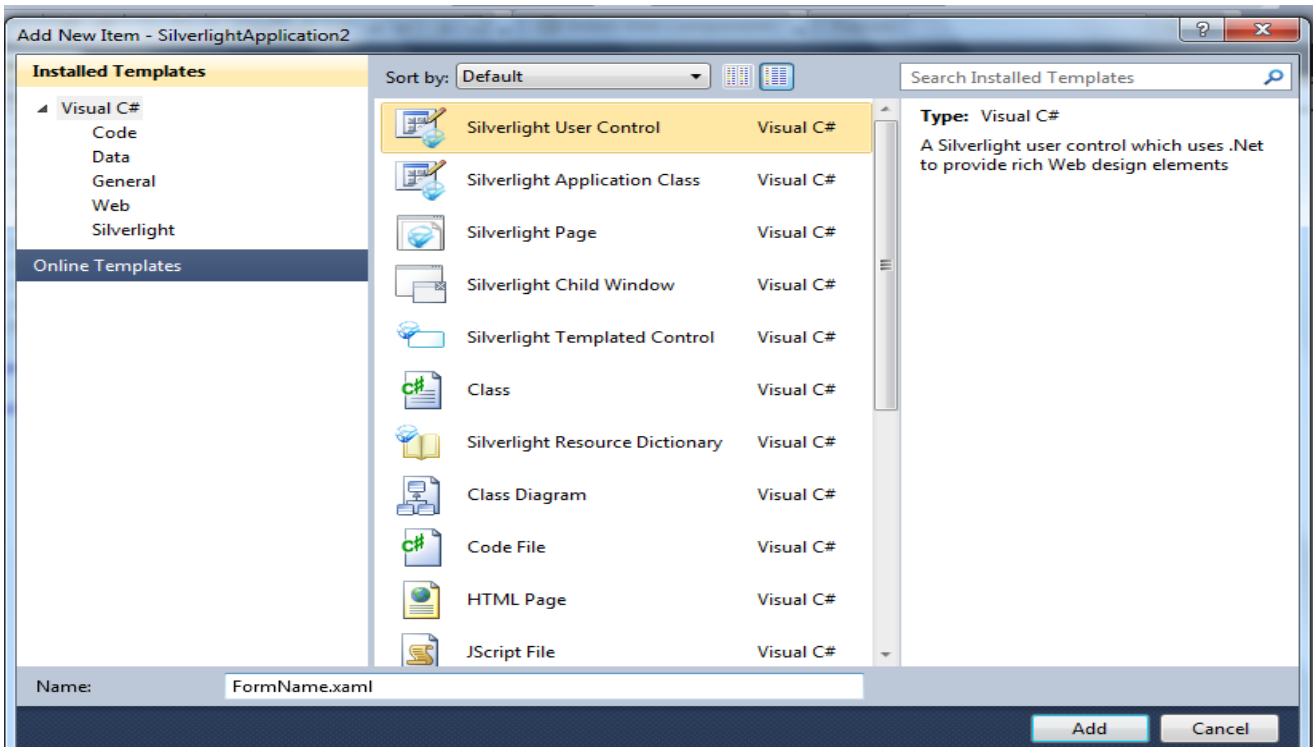
According to customer requirement, we can customize form and deliver the product according to specification. It has a lot of flexibility with the comparison of standard window of our product. We can use any kind of web controls or customize control.

2. Process To Create A Form

2.1 Step 1

In Silverlight, we create form in the Vienna Advantage AForms folder, if it's not there, better we create one. Then right click on AForms, ADD -> New Item ->Silver light User Control. Provide a name to the form accordingly as shown in below.



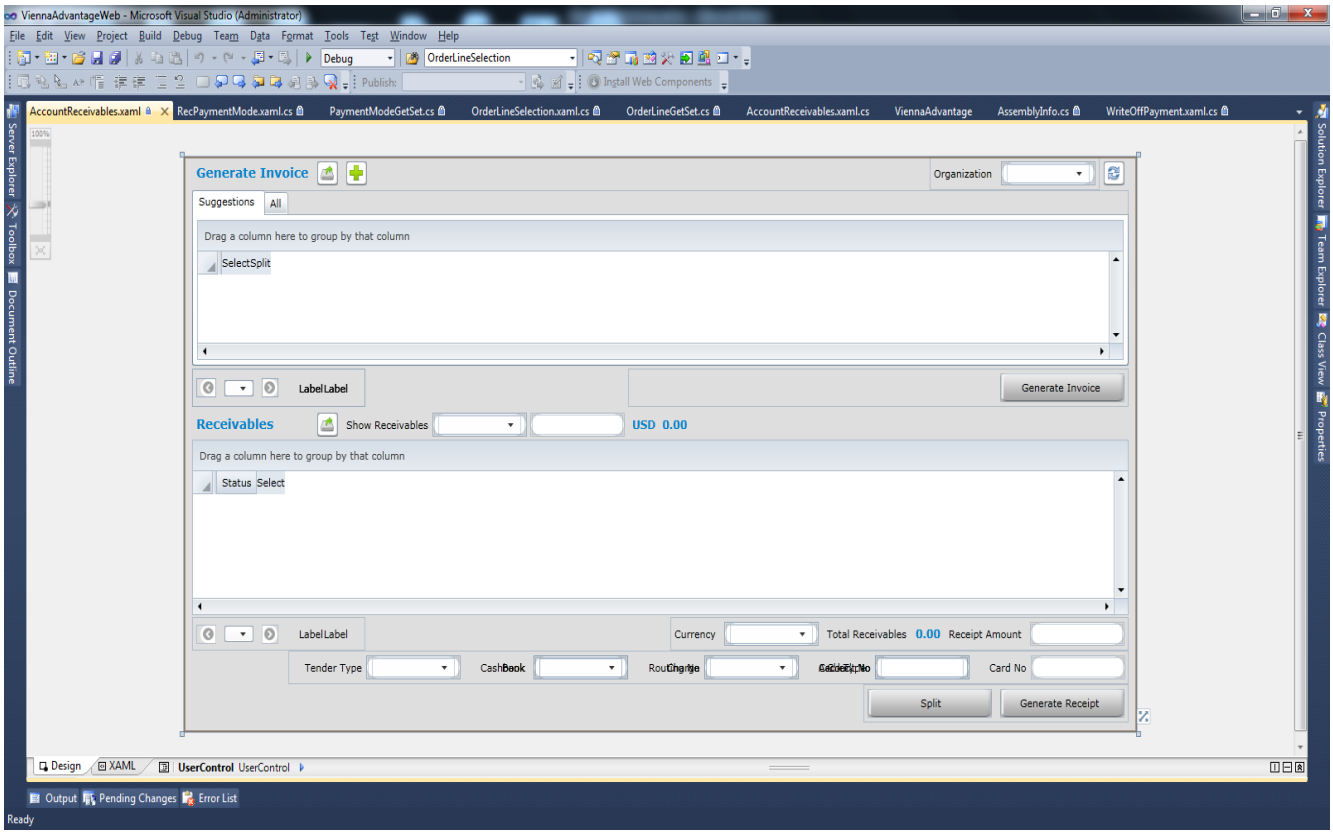


2.2 Step 2

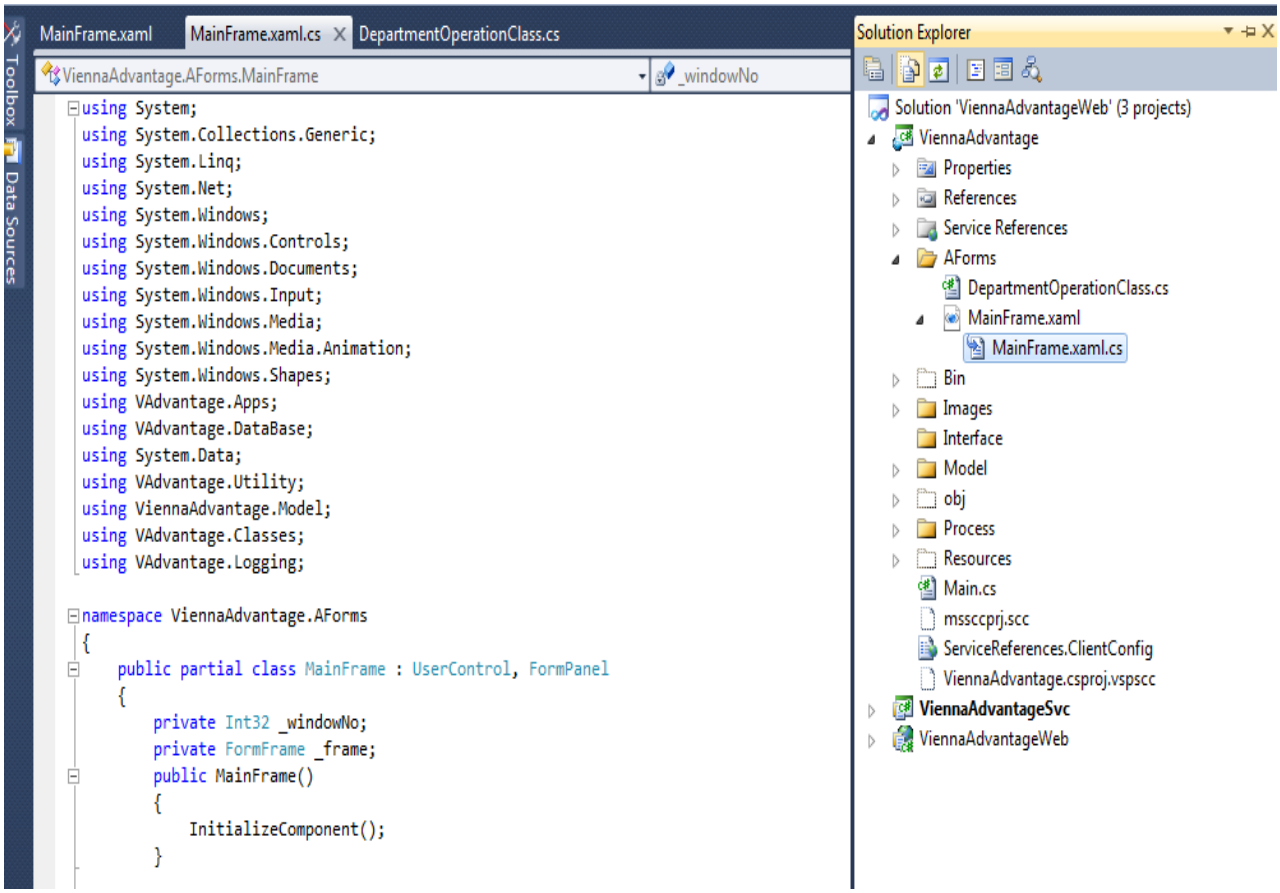
When We Add Form, there are two classes added in the AForms folder one with the extension .xaml and other as .xaml.cs.

- **.xaml**
- **.xaml.cs**

- **<u>.xaml</u> :-** .xaml File Contains Design of Form or you can say UI of the form. You can create design of the form by adding controls like grid, dropdowns and textboxes etc. which can be added from SLControls or Standard controls of Silverlight. To Add Controls just Double Click on .xaml file it will open designer of the form. Here you can design the form according to the requirement of the design. Below is the sample shown for the design of a form where a combobox, Textbox and grid is applied.



.xaml.cs :- .xaml.cs is the code behind of our form. All the events of the form are taken care in the .xaml.cs class. All the logics are implemented here regarding the form or you can implement a three tier architecture or MVVM to code your logics.

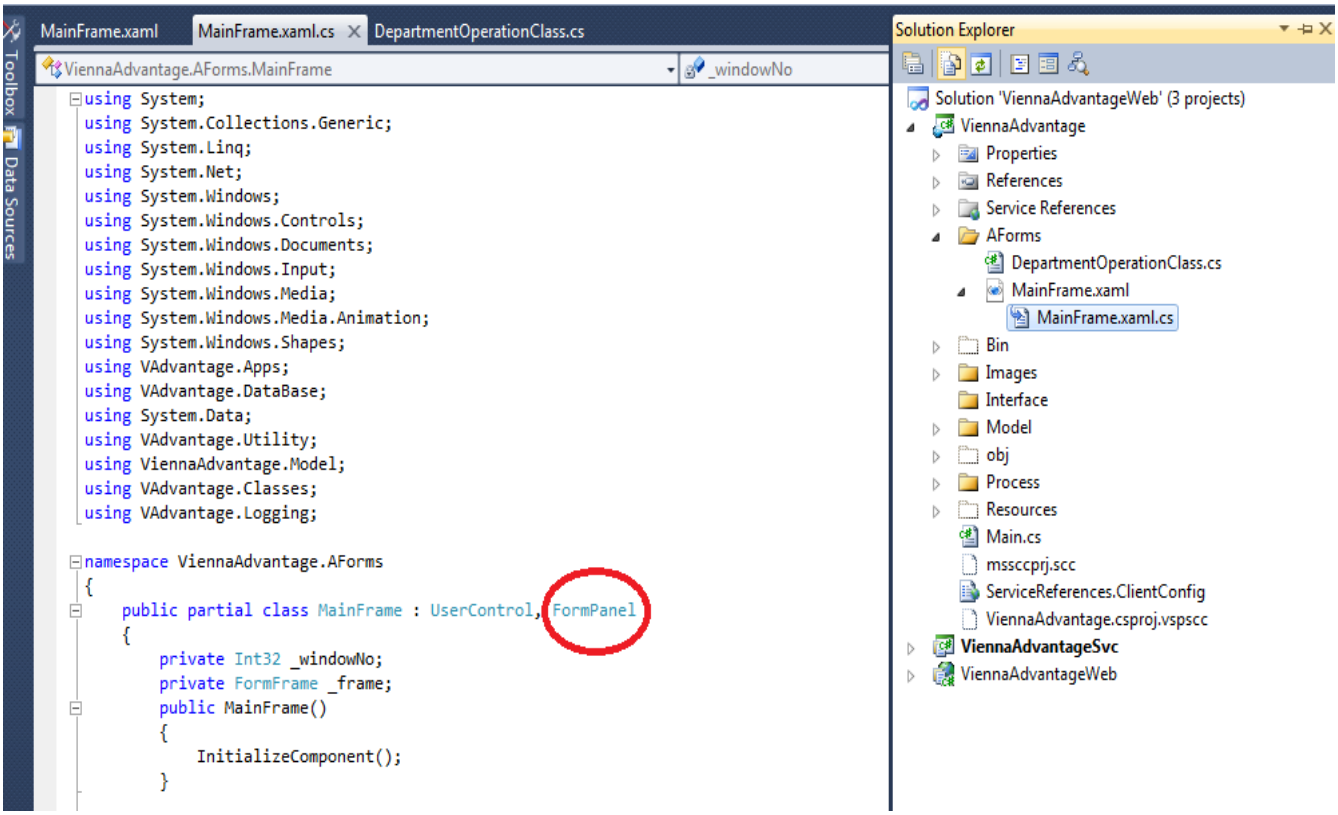


2.3 Step 3

Once the form is added and design is created next step is to Code the form. To start with we need to inherit the form from UserControl and FormPanel.

- **UserControl:-** User Control class is defined in **System.Windows.Controls** Namespace. Which provides classes to create elements, known as controls that enable a user to interact with an application.

- **FormPanel:-** FormPanel is an interface which is defined in **VAdvantage.Apps** Namespace. Now we need to implement FormPanel Interface which created two methods naming Init() and Dispose() which are explained further in the document.



Init():- It Initialize all the Components and after constructor is called this Init() Method Is Invoked. User can comment the default InitializeComponent() method from the default constructor and put in into Init(int WindowNo, FormFrame frame) method as show in below example or call the default InitializeComponent() method from default constructor. But make sure you do not put InitializeComponent() in both constructor and Init().

Dispose() :- It clears all the objects or set all the objects to null or we can say works like Garbage collector.

```
ViennaAdvantage.AForms.MainFrame
using VAdvantage.Utility;
using ViennaAdvantage.Model;
using VAdvantage.Classes;
using VAdvantage.Logging;

namespace ViennaAdvantage.AForms
{
    public partial class MainFrame : UserControl, FormPanel
    {
        private Int32 _windowNo;
        private FormFrame _frame;
        public MainFrame()
        {
            InitializeComponent();
        }


        public void Init(int windowNo, FormFrame frame)
        {
             InitializeComponent();
            _windowNo = windowNo;
            _frame = frame;
            _frame.ContentGrid.Children.Add(this);
            BindDepartment();
        }

        /// <summary> ...
        public void Dispose()
        {
            this.Dispose();
        }
```

2.4 Step 4

To Display design of a form at runtime we need to do the following :-

For .Net

frame.GetContentPane().Controls.Add((Control)this);

For Silver light

frame.ContentGrid.Children.Add(this);

3. Resize Design of User Control

To maximaze form in silverlight and .net these different methods are available.

For .Net

this.Dock = DockStyle.Fill;

Line use to resize the design of UserControl in the Form,

In Silverlight

```
try
{
   frame.FormState = SilverlightDesktop.Common.FormState.Maximized;
}
catch
{

}
```

4. Maintaining Logs on The form

Logs are way to keep track of Errors and warning information on the form. To maintain logs on the form one has to add namespace on the .xaml.cs class given below.

usingVAdvantage.Logging;

After adding the namespace an object of VLogger has to be initialized with name of the form as shown below, where **TestForm** is the name of the Form.

private static VLogger log = VLogger.GetVLogger(typeof(TestForm).FullName);

Save errors in log → **“log.SaveError("Amount Is Not Updated", "");”**

Where Amount Is Not Updated is the Message to be saved in the log file.

Save important information in log → **“log.Info("Currency=" + _C_Currency_ID); or log.SaveInfo(AdMessage,message);”**

Information saved in the log file will be like Currency = “ID of the currency”. In case where you want to save some information in the log file which is helpful when one checks the log file what went wrong in the form processing.

Some Errors which are severe can be put in the log file with Level.SEVERE in the log.

Catch exception→ **”log.Log(Level.SEVERE, "", ex);”** etc.

5. Get And Set Context

To set values in context or to fetch values from context Get and Set keywords are used. To work with this or use these we need to add a namespace VAdvantage.Utility.

Add namespace using **VAdvantage.Utility;**

Get login information from context e.g. get login clientID, Organisation id.

To get values from Context use :-

Envs.GetCtx().GetAD_Client_ID();

Envs.GetCtx().GetAD_Org_ID();

Envs.GetCtx().GetAD_Language(); etc.

To set values in context use :-

Envs.GetCtx().SetContext(_windowNo, "ColumnName", "Value");

Where _windowNo is the windowNo or tab opened which is provided in the Init() method of the form, and column Name and its value.

6. How To Show Message In Pop-Up Window

To show message in the popup to notify user we can use ShowMessage. And different options available are shown below. Where you provide the message KeyWordName as the name you want the user to be seen which will come from the message window.

ShowMessage.Warn("KeyWordName", null, "extraMessage", "ClassMethod");//Show warning

ShowMessage.Ask(-----);//Ask user

ShowMessage.Info(----);//show information

ShowMessage.Error(---);//show error.etc.

7. Get Text According to culture

We can make the form look and work according to the culture. And in order to do that you have to get the messages, labels and texts according to the culture. To get messages according to the culture we use Msg.GetMsg(Envs.GetCtx(), Message) method.

To show messages according to culture the namespace “using VAdvantage.Utility;” has to be added. Now when you want to get message according to culture we write like this

Msg.GetMsg(Envs.GetCtx(), Message) method.

Where Message is the keyword which it fetches from Message window according to the keyword provided. If no such message is found in the message window then it will show the text as given in the Message of the Msg.GetMsg method.

All the labels, buttons or checkboxes etc. text should come from Message window so that culturing can be implemented in the form. And to do so we need to use GetMsg() method of Msg classs.

8. Working with Database

In order to work with Database, we need to understand an important concept of Threading. There are two threads which we need to handle in form code, one is Child Thread and the other is Main or Parent Thread.

In child Thread all the Queries like Select, Insert, Update are executed. You can't access any form controls in child thread, it will give error, or result in processing. So make sure you don't access form controls in child thread.

Parent Thread or Main Thread is responsible to handle Controls. All controls are accessed in the main thread. You can't execute any queries in Main thread. Important thing to note here is, we can't make Parent Thread or Main thread to wait.

In the images below you can see how to work with threading. To execute a query we have to execute it in Child or second Thread. Important to note here to execute a query we use **DB class .** To work with DB class we need to add a namespace **"using VAdvantage.Database;".** There are different methods in DB Class as there are in standard ADO like ExecuteDataset, ExecuteReader etc., e.g. DB.ExecuteDataset is used in the image below.

In the Main Thread we use to get and set values e.g. in combobox1 control an itemsource is provided to the Drop down list.

```
public void BindDepartment()
{
    string query = string.Empty;
    DataSet ds = null;
    //second thread
    System.Threading.ThreadPool.QueueUserWorkItem(delegate
    {
        query = @"select trn4_departmentname,trn4_department_id from trn4_department order by trn4_department_id";
        ds = new DataSet();
        try
        {
            ds = DB.ExecuteDataset(query, null, null);
        }
        catch (Exception ex)
        {

        }
        //first thread
        Dispatcher.BeginInvoke(delegate
        {
            if (ds != null)
            {
                if (ds.Tables[0].Rows.Count > 0)
                {
                    comboBox1.ItemsSource = ds.Tables[0].GetBindableData(new Connector());
                    comboBox1.DisplayMemberPath = "TRN4_DEPARTMENTNAME";
                    comboBox1.SelectedValuePath = "TRN4_DEPARTMENT_ID";
                }
            }
        });
    });
}
```

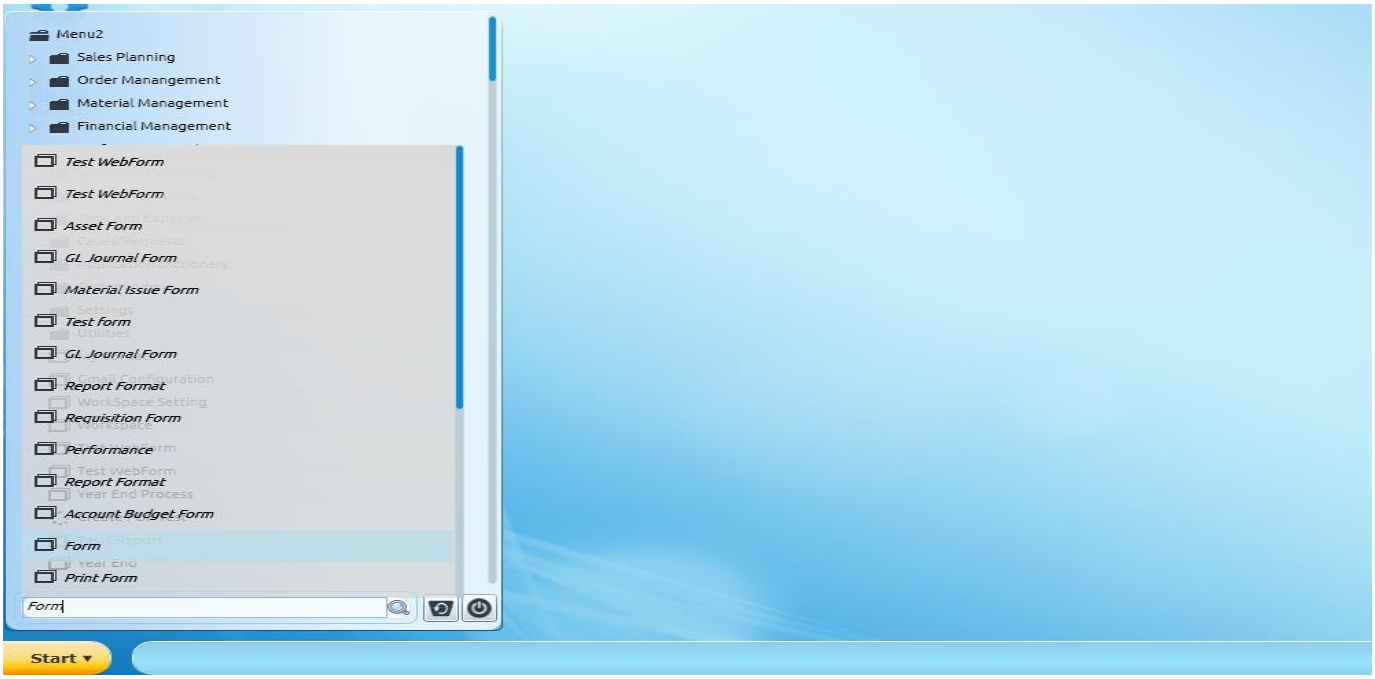
Second Thread (To Access DB)

Main Thread

When, Second Thread Completes its Job (**Get or Set the Data From/To the Database**) then, it comes to Main Thread.

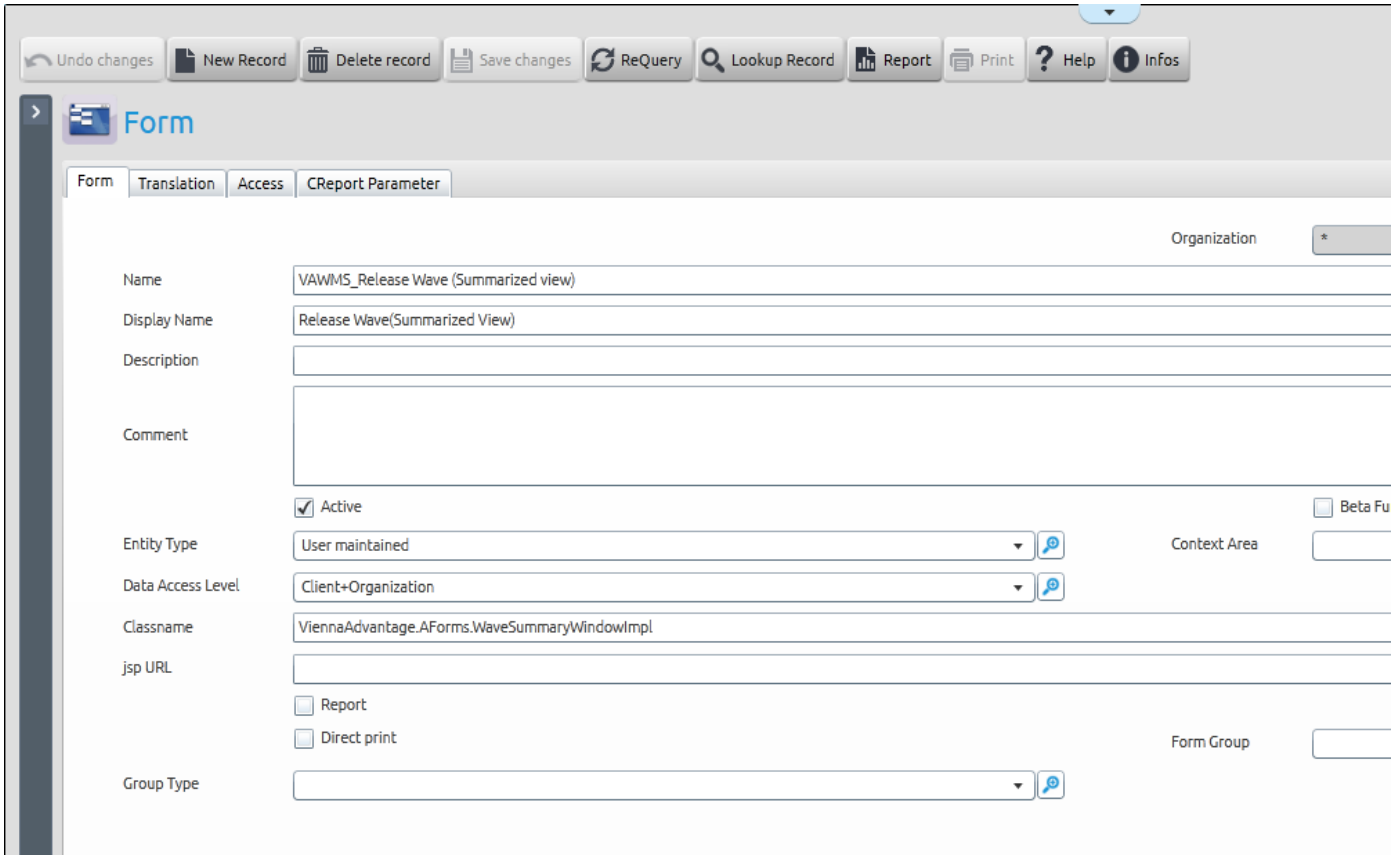
9 Entry in the Form Window

Now when the coding part is over we have to make some entries in few windows to be able to see the form in the menu. To do so first step is to make the entry in Form Window. Go to Start in Vienna Advantage Taskbar and type Form in the search you will find a menu as shown below.



It will open Form window where you need to enter the details of the form like Name of the form (In case of form created for Module it should be with prefix), Display name for the form, Description, comments, Entity Type, Data Access Level and ClassName etc.

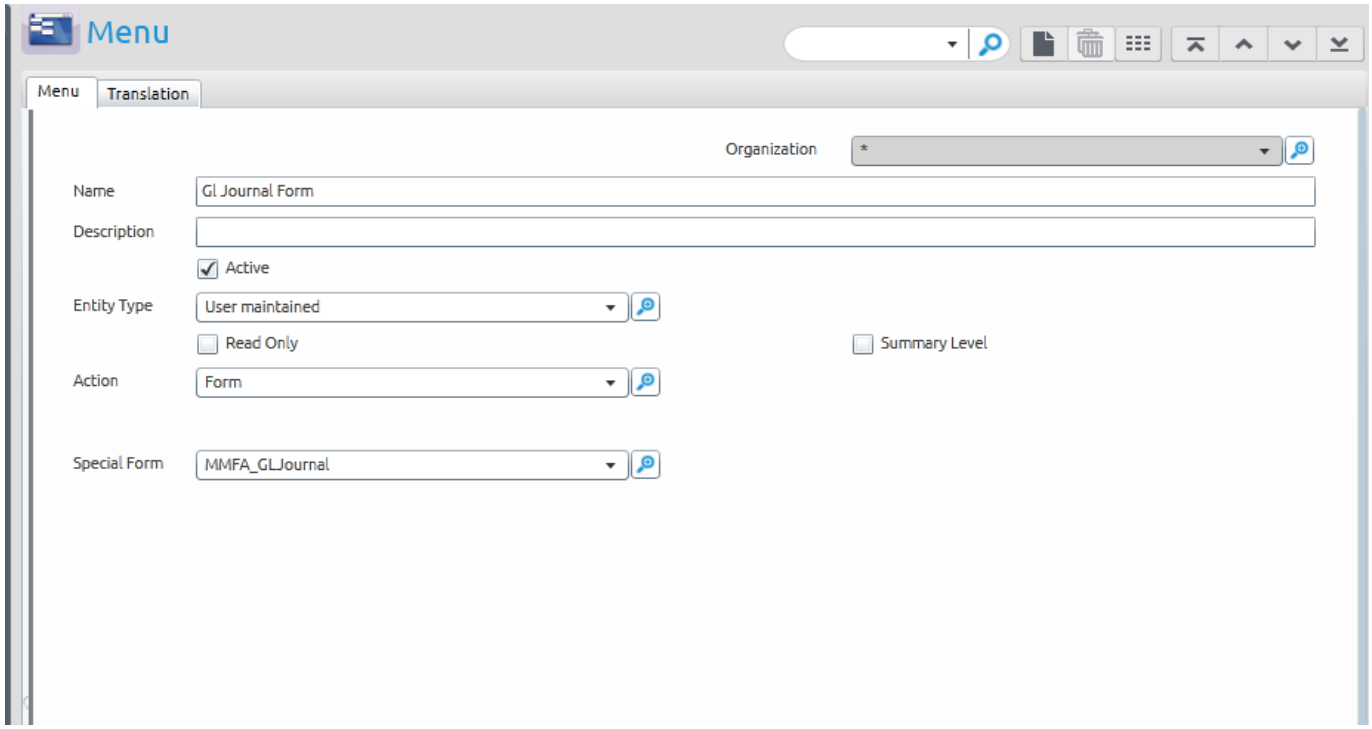
ClassName field in Form window is the name of the form which is created in the AForms, and it should be with the namespace of the form e.g. in the image below **"ViennaAdvantage.AForms.WaveSummaryWindowImpl"** as it looks for the form with this name. If there is something wrong with the classname field it will not load the form.



10 Bind Form with Menu

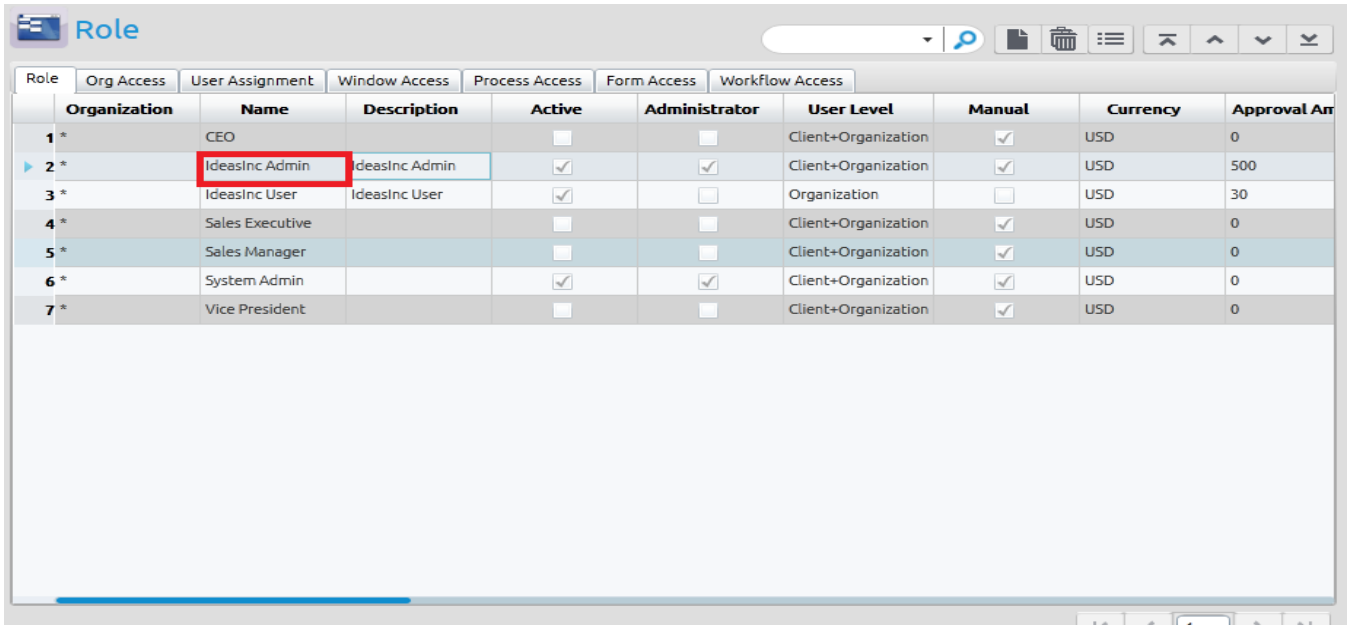
Now to show form in the menu we have to bind form with the menu. In order to do so open menu window from Vienna Advantage Taskbar. Go to Start and type menu in the search. You will find menu window in the search list and select menu as shown below.

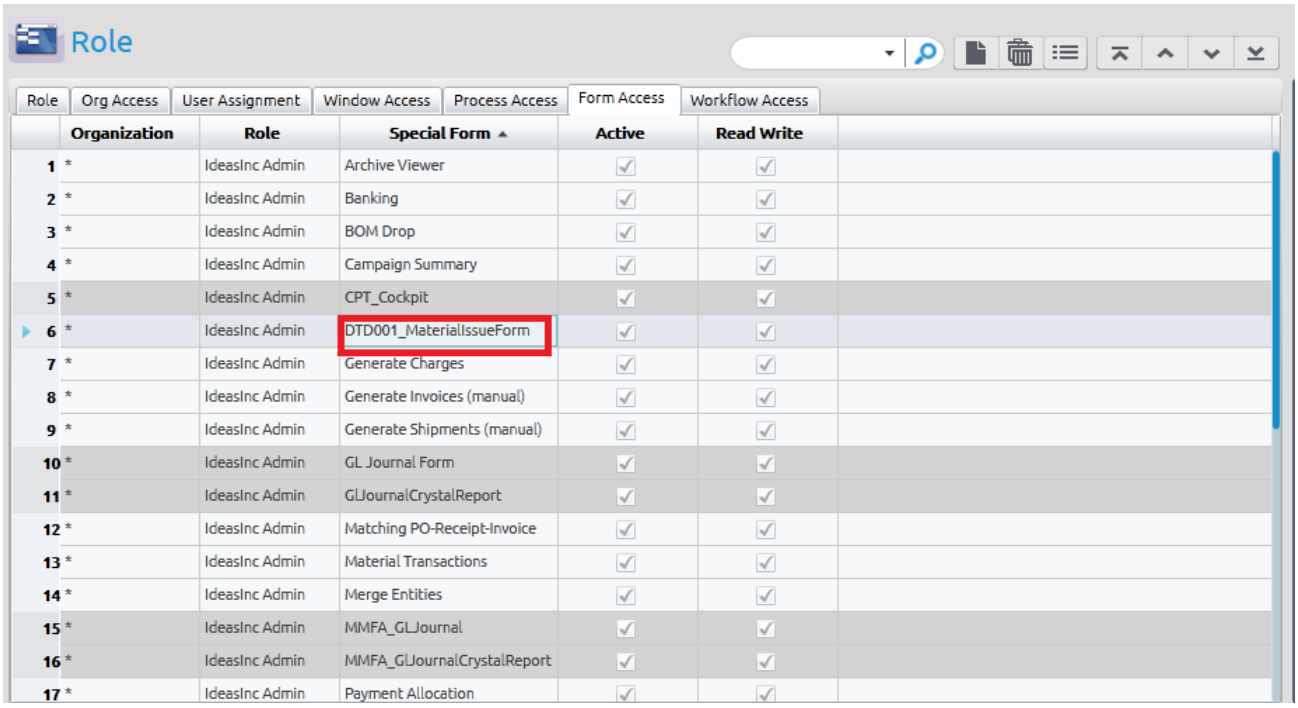
Now in the Menu window provide the name you for the form you want to see in the menu. Description for the menu. Select action as Form in the action dropdown, which will enable Special Form field. And select form which is created in the special form field.



11 Give Access to Form

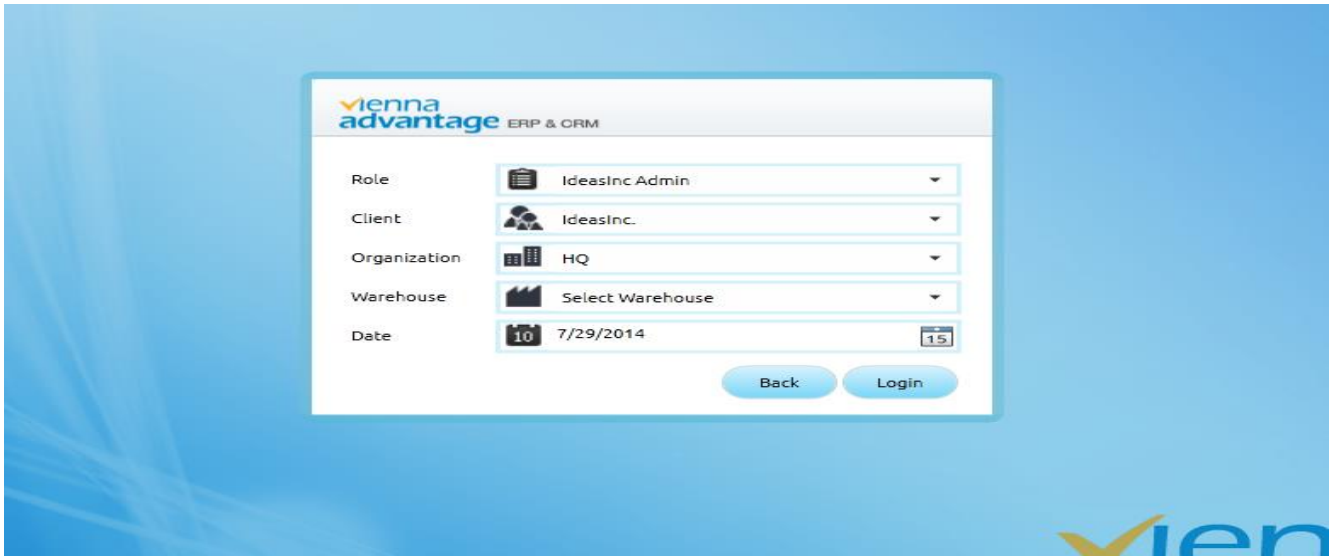
After this open role window from menu to whom you want to give access of this form. In role window select the role and go to Form Access Tab. Check Read/Write checkbox in the Form access tab if form is already there otherwise make a new entry of the form and check Read/Write to true as shown in the image below.




Role
Role
Org Access
User Assignment
Window Access
Process Access
Form Access
Workflow Access
Organization
Role
Special Form
Active
Read Write
1 * IdeasInc Admin Archive Viewer
2 * IdeasInc Admin Banking
3 * IdeasInc Admin BOM Drop
4 * IdeasInc Admin Campaign Summary
5 * IdeasInc Admin CPT_Cockpit
6 * IdeasInc Admin DTD001_MaterialIssueForm
7 * IdeasInc Admin Generate Charges
8 * IdeasInc Admin Generate Invoices (manual)
9 * IdeasInc Admin Generate Shipments (manual)
10 * IdeasInc Admin GL Journal Form
11 * IdeasInc Admin GLJournalCrystalReport
12 * IdeasInc Admin Matching PO-Receipt-Invoice
13 * IdeasInc Admin Material Transactions
14 * IdeasInc Admin Merge Entities
15 * IdeasInc Admin MMFA_GLJournal
16 * IdeasInc Admin MMFA_GLJournalCrystalReport
17 * IdeasInc Admin Payment Allocation

12 Run The Form

Now login with the role for which you have given the access of that form e.g. given the role to IdeasIncAdmin so login with IdeasIncAdmin role.



Look for the Form which you added in the Menu window. Go to Start in the Vienna Advantage Taskbar and type name of the form in the search field, the form which you created can be seen in the search result as shown below.

Now click on the Menu item or the form name, you are now able to use the form.